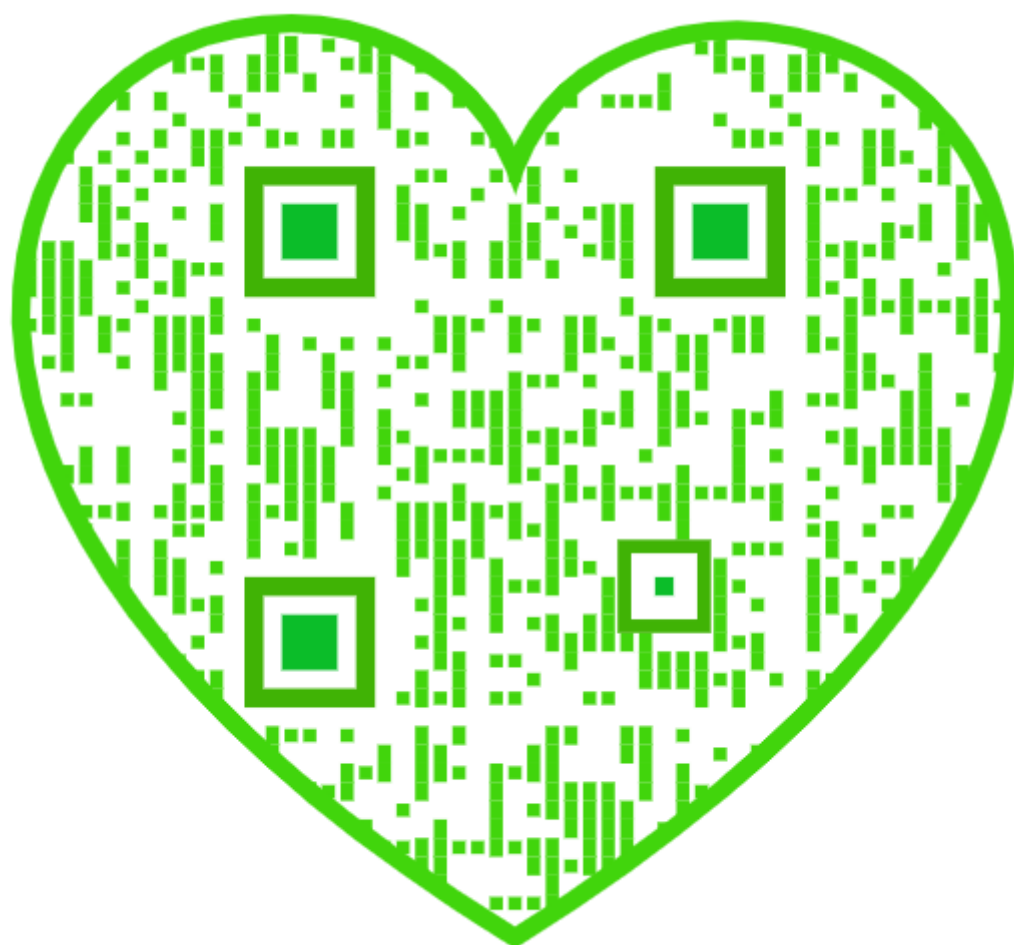


# Master in Artificial Intelligence



## Algorithm Selection & Development XVIII







# Purpose

**The purpose of the section is to help you learn how to research, select, and develop appropriate algorithms to become a Successful Artificial Intelligence (AI) Engineer**

**At the end of this lecture, you will learn the following**

**How to train Gradient boosting machines algorithm for getting feature importance**



# How to train Gradient boosting machines algorithm for getting feature importance

Decision  
trees

Random  
forests

Gradient  
boosting  
machines



# Using XGBoost

## Import Libraries

python

```
import xgboost as xgb  
from sklearn.datasets import load_iris
```



# Using XGBoost

## Load Dataset

```
python

# Load sample dataset (you can replace it with your own dataset)
data = load_iris()
X = data.data # Features
y = data.target # Target variable
```



# Using XGBoost

## Train XGBoost Model

python

```
# Define parameters
params = {
    'objective': 'multi:softmax', # for classification
    'num_class': 3, # number of classes
    'max_depth': 3 # maximum depth of trees
}

# Initialize XGBoost classifier
xgb_clf = xgb.XGBClassifier(**params)

# Fit the model
xgb_clf.fit(X, y)
```



# Using XGBoost

## Get Feature Importance Scores

python

```
# Extract feature importances
feature_importances = xgb_clf.feature_importances_

# Print or visualize feature importances
for i, importance in enumerate(feature_importances):
    print(f"Feature {i}: {importance}")
```





# Using scikit-learn GradientBoostingClassifier

## Import Libraries

python

```
from sklearn.datasets import load_iris  
from sklearn.ensemble import GradientBoostingClassifier
```



# Using scikit-learn GradientBoostingClassifier

## Load Dataset

python

```
# Load sample dataset (you can replace it with your own dataset)
data = load_iris()
X = data.data # Features
y = data.target # Target variable
```



# Using scikit-learn GradientBoostingClassifier

## Train Gradient Boosting Model

python

```
# Initialize Gradient Boosting classifier
gb_clf = GradientBoostingClassifier()

# Fit the model
gb_clf.fit(X, y)
```



# Using scikit-learn GradientBoostingClassifier

## Get Feature Importance Scores

python

```
# Extract feature importances
feature_importances = gb_clf.feature_importances_

# Print or visualize feature importances
for i, importance in enumerate(feature_importances):
    print(f"Feature {i}: {importance}")
```





# How to train Gradient boosting machines algorithm for getting feature importance

Decision  
trees

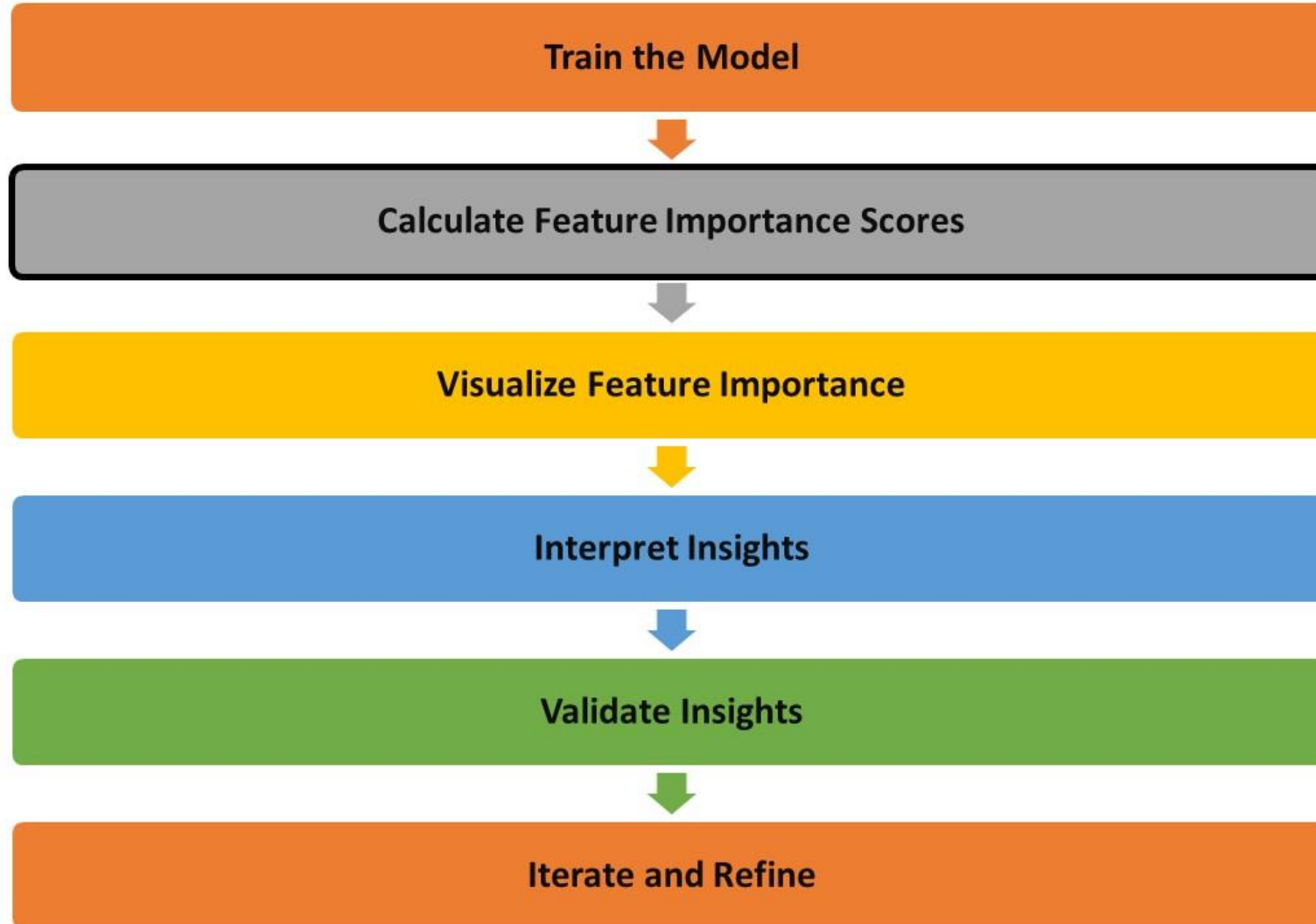
Random  
forests

Gradient  
boosting  
machines



# What is next?

How to use feature importance analysis to provide insights into model predictions





# Master in Artificial Intelligence



## Algorithm Selection & Development XVIII

